Bayesian Optimization of Combinatorial Structures

Supplementary Material

A. General Form of BOCS

In this section we describe the general form of BOCS that handles categorical and integer-valued variables and models of order larger than two.

So far, we have focused our presentation on binary variables, i.e., $\mathcal{D} = \{0,1\}^d$ or equivalently, $\mathcal{D} = \{-1,+1\}^d$, that allow efficient encodings of many combinatorial structures as demonstrated above.

We begin with a description of how to incorporate categorical variables into our statistical model. Let $\mathcal{I}$ denote the indices of categorical variables. Consider a categorical variable x_i with $i \in \mathcal{I}$ that takes values in $\mathcal{D}_i = \{e_1^i, \dots, e_{m_i}^i\}$. We introduce m_i new binary variables x_{ij} with $x_{ij} = 1$ if $x_i = e_j^i$ and $x_{ij} = 0$ otherwise. Note that $\sum_j x_{ij} = 1$ for all $i \in \mathcal{I}$ since the variable takes exactly one value, and the dimensionality of the binary variables increases from d to $d - |\mathcal{I}| + \sum_{i \in \mathcal{I}} m_i$.

BOCS uses the sparse Bayesian linear regression model for binary variables proposed in Sect. 3.1 and samples $\boldsymbol{\alpha}_t$ in each iteration t. When searching for the next $x^{(t)}$ that optimizes the objective value for $\boldsymbol{\alpha}_t$, we need to exert additional care: observe that running SA to optimize the binary variables might result in a solution that selects more than one element in $\mathcal{D}_i$, or none at all, and therefore would not correspond to a feasible assignment to the categorical variable x_i.

Instead, we undo the above expansion: SA operates on d-tuples x where each x_i with $i \in \mathcal{I}$ takes values in its original domain $\mathcal{D}_i$. Then the neighborhood $N(x)$ of any tuple x is given by all vectors where at most one variable differs in its assignment. To evaluate $f_{\boldsymbol{\alpha}_t}(x) + \mathcal{P}(x)$, we leverage this correspondence between categorical variables and their 'binary representation'.

Note that integer-valued variables can be handled naturally by the regression model. For the optimization of the acquisition criterion, simulated annealing uses the same definition of the neighborhood $N(x)$ as in the case of categorical variables.

Next we show how to extend the BOCS algorithm to models that contain monomials of length larger than two. In this case we have

$$f_\alpha(x) = \sum_{S \in 2^{\mathcal{D}}} \alpha_S \prod_{i \in S} x_i,$$

where $2^{\mathcal{D}}$ denotes the power set of the domain and α_S is a real-valued coefficient.

Following the description of BOCS for second-order models, the regression model is obtained by applying the sparsity-inducing prior described in Sect. 3.1. Then, in each iteration t, we sample $\boldsymbol{\alpha}_t$ from the posterior over the regression coefficients and now find

$$x^{(t)} \in \operatorname*{argmax}_{x \in \mathcal{D}} \sum_{S \in 2^{\mathcal{D}}} \alpha_S \prod_{i \in S} x_i + \mathcal{P}(x).$$

Since we can evaluate the objective value $f_{\boldsymbol{\alpha}_t}(x) + \mathcal{P}(x)$ at any x efficiently, we may use simulated annealing again to search for an optimizer of the acquisition criterion.

B. Evaluation of Higher Order Models

We point out that the problems studied in Sect. 4.2, 4.3, and 4.4 have natural interactions of order higher than two between the elements that we optimize over. To highlight these interactions, we measure the number of regression coefficients that have significant weight (i.e., values $|\alpha_i| \geq 0.1$) with the sparse regression model of different orders.

As an example, we fit the model using 100 samples from a random instance of the Ising model presented in Sect. 4.2. Typically, four out of 24 linear terms, 28 out of 300 second-order terms, and 167 out of 2048 third-order terms have value of at least 0.1. Here we note the importance of the sparsity-inducing prior to promote a small number of parameters in order to reduce the variance in the model predictions (cp. Sect. 3.1).

We also examine how BOCS performs when equipped with a statistical model of higher order. Our implementation follows Sect. 3.1 and uses simulated annealing to search for an optimizer of the acquisition criterion as described in Sect. A and in Sect. 3.4.

Fig. 8 compares the performances of the BOCS-SA algorithm on the aero-structural benchmark with a second and third-order model. The second-order model has a lower number of coefficients that can be estimated with lower variance given few training samples. On the other hand, a statistical model of higher order is able to capture more interactions between the active coupling variables but may require more samples for a sufficient model fit. Thus, it is not surprising that BOCS-SA performs better initially with the second-order model. As the number of samples grows larger, the third order model obtains better results.

We also evaluate BOCS-SA with higher order models for the Ising benchmark presented in Sect. 4.2. Fig. 9 contrasts the performances of the BOCS-SA algorithm for $\lambda = 0$ with a first order, a second-order, and a third-order statistical model. All results are averaged over 100 instances of the Ising model. Fig. 10 summarizes the results for $\lambda = 10^{-2}$. Interestingly, the third-order model already performs similarly to the second-order model for this problem with a

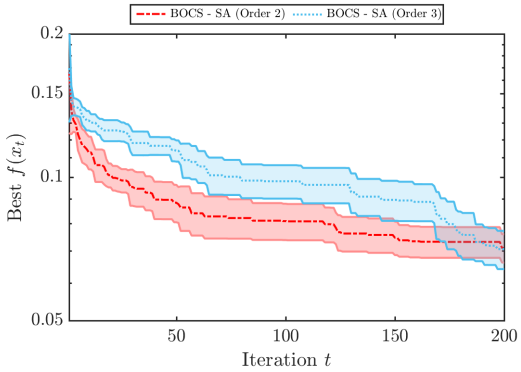

Figure 8. Performance of BOCS−SA on the aero-structural benchmark for $\lambda = 10^{-2}$ with second and third-order statistical models. As the number of samples increases, BOCS−SA with the third-order model achieves better results.

smaller number of data points, although it exhibits a larger variability in the performance.

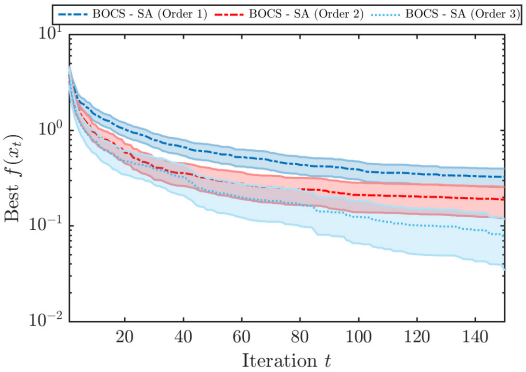

Figure 9. Performance of BOCS−SA for the Ising model with λ=0. After 150 iterations, BOCS−SA with the third order model performs better on average.

These results provide numerical evidence that a second-order model provides a good trade-off of model expressiveness and accuracy for these problems when data is limited.

C. Wall-clock time Performance

In this section, we compare the wall-clock times required by BOCS and EI for the Ising benchmark presented in Sect. 4.2. The wall-clock time is computed as the first time each instance of the algorithm reaches an objective value of 0.01 for $\lambda = 0$, and 10^{-4}. The average results over 100 runs of each algorithm and the 95% confidence intervals are presented in Table 4.

The results demonstrate that BOCS is considerably faster

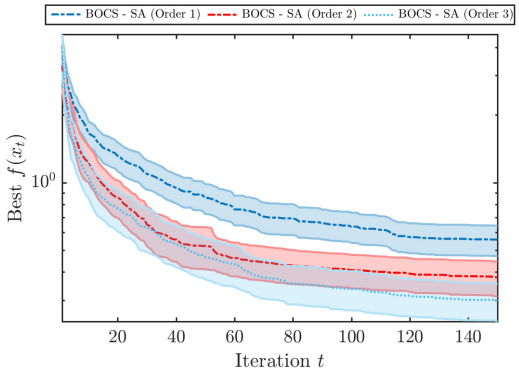

Figure 10. Performance of BOCS−SA on the Ising model with λ=10^{-2}. After 150 iterations, the third order models typically leads to better results.

Table 4. Wall-clock time required by the algorithms presented in Sect. 3 for the 24-dimensional Ising model benchmark. BOCS−SA and BOCS−SDP are considerably more efficient than EI.

λ	EI	BOCS−SA	BOCS−SDP
0	404.2 ± 49.1	**45.8** ± 12.4	115.6 ± 18.1
10^{-4}	412.9 ± 55.8	**62.1** ± 16.5	104.6 ± 16.7

than EI. BOCS−SA is at least seven times faster, while BOCS−SDP is still four times faster. Even for a problem with 24 binary variables, the cost of finding an optimizer of the acquisition function is prohibitively large for EI.

D. Descriptions of Benchmark Problems

In this section, we provide more details on the benchmark problems studied in Sect. 4.

D.1. Sparsification of Ising Models

To evaluate the objective function for the Ising model (see Sect. 4.2), we compute the KL divergence between models $p(z)$ and $q_x(z)$, that are defined by their interaction matrices J^p and J^q, respectively. To do so, we pre-compute the second moments of the random variables in the original model given by $\mathbb{E}_p[z_i z_j]$ and use these together with the differences in the interaction matrices to evaluate the first term in the KL divergence. The second term in the objective is given by the log difference of the partition functions, $\log(Z_q/Z_p)$, where Z_p is constant for each x and only needs to be evaluated once. Z_q is the normalizing constant for the approximating distribution and is given by

$$Z_q = \sum_{z \in \{-1,1\}^n} \exp(z^T J^q z). \tag{8}$$

In this work we do not restrict the class of distributions to be defined over subgraphs $\mathcal{G}^q$ whose normalizing constants can

be computed efficiently (e.g., mean-field approximations). Therefore, in general, computing Z_q requires summing an exponential number of terms with respect to n, making this term and the KL divergence expensive to evaluate.

Furthermore, the objective function above only measures the distance between $p(z)$ and an approximating distribution defined over a subgraph while still using the same parameters, i.e., if J^q_{ij} is non-zero then it has the same value as the corresponding entry in J^p.

D.2. Contamination Control

The objective in this problem is to minimize the cost of prevention efforts while asserting that the contamination level does not exceed certain thresholds with sufficiently high probability. These latter constraints are evaluated by running T Monte Carlo simulations and counting the number of runs that exceed the specified upper limits for the contamination. Each set of Monte Carlo runs defines an instance of the objective function in Eq. (7) that we optimize with respect to x using the various optimization algorithms presented in Sect. 3. In our studies we followed the recommended problem parameters that are provided by the `SimOpt` Library (Hu et al., 2010).

D.3. Aero-structural Multi-Component Problem

The *OpenAeroStruct* model developed by (Jasa et al., 2018) computes three output variables for each set of random input parameters. In this study, our objective is to evaluate the change in the probability distribution of these outputs for each set of active coupling variables between the components of the computational model, x. We denote the distribution of the outputs in the reference model by $\pi_{\mathbf{y}}$ (i.e., with all active coupling variables) and the decoupled model by $\pi^x_{\mathbf{y}}$. The difference in these probability distributions is measured by the KL divergence and is denoted by $D_{KL}(\pi_{\mathbf{y}}||\pi^x_{\mathbf{y}})$.

While the KL divergence can be estimated to arbitrary accuracy with Monte Carlo simulation and density estimation techniques, in this study we follow Baptista et al. (2018) and rely on an approximation of the objective. This approximation linearizes the components of the model and propagates the uncertainty in the Gaussian distributed input variables to characterize the Gaussian distribution for the outputs. By repeating this process for the reference and decoupled models, an estimate for the KL divergence can be computed in closed form between the two multivariate Gaussian distributions. However, the linearization process still requires computing gradients with respect to high-dimensional internal state variables within the model and is thus computationally expensive.

For more information on how to evaluate the approximate KL divergence as well as its numerical performance in practice for several engineering problems, we refer the reader to Baptista et al. (2018).

E. Maximum Likelihood Estimate for the Regression Coefficients

In Sect. 3.1 we proposed a Bayesian treatment of the regression coefficients $\boldsymbol{\alpha}$ in Eq. (1). Here we discuss an alternative approach based on a point-estimate, e.g., a maximum likelihood estimate (MLE). Suppose that we have observed $(x^{(i)}, f(x^{(i)}))$ for $i = 1, \ldots, N$. The maximum likelihood estimator assumes that the discrepancy between $f(x)$ and the statistical model is represented with an additive error. This error is supposed to follow a normal distribution with mean zero and known finite variance σ^2. To compute this estimator, we stack the p predictors of all N samples to obtain $\mathbf{X} \in \{0, 1\}^{N \times p}$. Then the regression coefficients $\boldsymbol{\alpha}$ are obtained by the least-squares estimator

$$\boldsymbol{\alpha}_{MLE} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y}), \tag{9}$$

where $\mathbf{y} \in \mathbb{R}^N$ is the vector of function evaluations.

While the parameters of this model can be efficiently estimated for a small number of evaluations, the MLE only provides a uniform estimate of σ^2 for the variance of the coefficients. On the other hand, the Bayesian models described in Section 3.1 better characterize the joint uncertainty of all parameters $\boldsymbol{\alpha}$ and σ^2 in order to capture the discrepancy of the generative model. `BOCS` leverages this uncertainty by sampling from the posterior distribution over the coefficients. This sampling allows `BOCS` to better explore the combinatorial space of models and find a global optimum of the objective function. This is also contrasted with using a variance of σ^2 to sample the coefficients independently, which may lead to uninformative models that do not account for the correlation between coefficients that is captured by the Bayesian models. Furthermore, we note that only using the MLE coefficients from (9) in `BOCS` often results in purely exploitative behavior that fails to find the global optimum, as observed in Fig. 2.

F. Validation of the Regression Models

We now validate models of order two proposed in Sect. 3.1 and Sect. E for each benchmark considered in Sect. 4. The figures compare the statistical models based on the maximum likelihood estimate, standard Bayesian linear regression and sparse Bayesian linear regression based on the sparsity-inducing prior introduced in Sect. 3.1. The standard Bayesian linear regression model supposes a joint prior for the parameters of $P(\boldsymbol{\alpha}, \sigma^2) = P(\boldsymbol{\alpha}|\sigma^2)P(\sigma^2)$, where $\boldsymbol{\alpha}|\sigma^2 \sim \mathcal{N}(\mu_\alpha, \sigma^2\Sigma_\alpha)$ and $\sigma^2 \sim IG(a, b)$ have a normal and inverse-gamma distribution respectively. Given the

same data model as in Sect. 3.1, the joint posterior of $\boldsymbol{\alpha}$ and σ^2 has a normal-inverse-gamma form.

We compare the average absolute approximation error of these three regression models on a test set of $M = 50$ points, varying the number of training points.

F.1. Validation on Binary Quadratic Programming.

We first evaluate the regression models on an instance of the test function from Sect. 4.1, using a set of $N = 40$ training samples. Fig. 11 depicts: the true function values (black), the predictions of the MLE estimator (red), and the mean and standard deviation of the Bayesian linear regression model (green) and of the sparse regression model (blue). The regression model with the sparsity-inducing prior (blue) achieves the best prediction of the true values (black). This figure empirically demonstrates the ability of

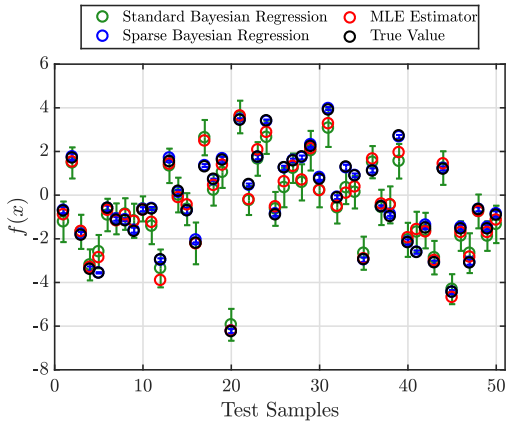

Figure 11. Test error of quadratic test problem ($L_c = 1$) for different $\boldsymbol{\alpha}$ estimators. The Bayesian regression with a sparsity-inducing prior (blue) performs better than the Bayesian linear regression model (green) and the MLE estimator (red).

the model to accurately capture the effect of binary coupling between input variables. Although the MLE also provides good estimates, as we discuss in Sect. 4, the performance of the Bayesian optimization process is drastically impaired when using the MLE estimator instead of samples from the posterior of the regression coefficients, since the uncertainty in the model is not reflected in the former.

We now compare the average test error of $M = 50$ points with an increasing size of the training set in Fig. 12. The results are averaged over 30 random instances of the binary quadratic problem (BQP) with $L_c \in [1, 10, 100]$. As N increases, the test error is converging for all estimators. We note that for a quadratic objective function, the quadratic model $f_{\boldsymbol{\alpha}}(x)$ closely interpolates the function with a sufficient number of training points, resulting in low test error for the MLE estimator. We note that for this lower $d = 10$-dimensional test problem, standard Bayesian linear regression (green) resulted in similar accuracy as the sparse estimator (blue).

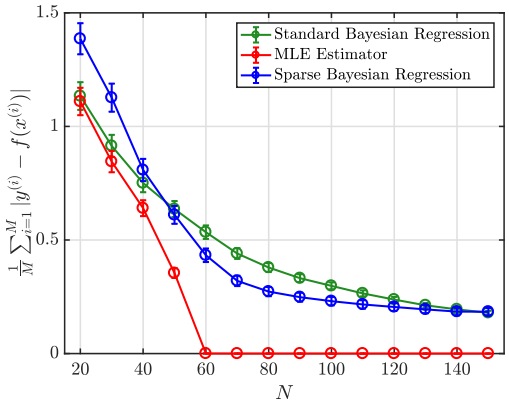

Figure 12. Test error of quadratic test problem with increasing size of training set. Standard Bayesian regression (green) and sparse regression (blue) perform similarly as N increases for the $d = 10$ quadratic test problem.

F.2. Validation on the Ising Problem.

For the Ising model with $d = 24$ edges, we examine the test error of $M = 50$ points with an increasing size of the training set; see Fig. 13. The results are averaged over 10 models with randomly drawn edge weights as discussed in Section 4.2 and the 95% confidence intervals of the mean error are also reported in the error bars. As compared to the results for the BQP, the sparse estimator provides lower test errors for this higher-dimensional problem, warranting its use over Bayesian linear regression in the `BOCS` algorithm. This reduction in test error can be attributed to the shrinkage of coefficients with near-zero values from the sparsity-inducing prior (Carvalho et al., 2010).

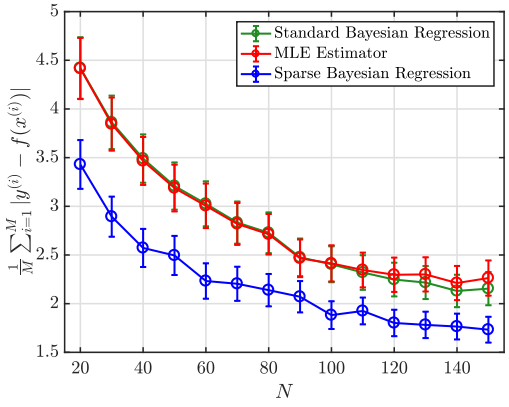

Figure 13. Test error of Ising model for different $\boldsymbol{\alpha}$ estimators. The sparse estimator (blue) provides lower errors on the test set than standard Bayesian linear regression (red).

F.3. Validation on the Contamination Control Problem.

The test error with increasing training set size is plotted in Fig. 14 for the contamination control problem with $d = 20$

stages and $T = 10^3$ Monte Carlo samples for approximating the probability in the objective.

With increasing N, the variance in the values of all estimated coefficients decreases, which results in lower test set error as observed for the MLE and Bayesian linear regression. A similar behavior is also seen for the sparse estimator with a large reduction in the error offset for small values of N. This suggests that the objective can be well approximated by the model described in Section 3.1 with a sparse set of interaction terms. As a result, the sparsity-inducing prior learns the set of non-zero terms and the test set error is dominated by the variance of the few remaining terms. It seems advantageous for BOCS to have a more accurate model based on this sparse prior when N is small.

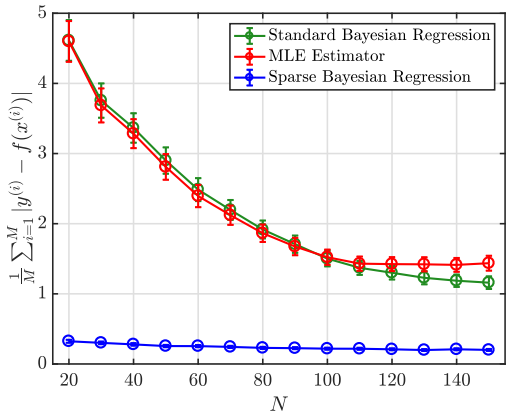

Figure 14. Test error of contamination control problem for different $\boldsymbol{\alpha}$ estimators. The model based on the sparsity-inducing prior (blue) provides the best performance for approximating the objective.

F.4. Validation on the Aero-structural Problem.

For the aero-structural problem in Section 4.4, the average absolute test set error for $M = 50$ samples is presented in Fig. 15. With an increasing number of training samples, this problem has similar performance for the four different estimators. We note that for large N, the test set error of the $\boldsymbol{\alpha}$ estimators for this problem begin to plateau with more training samples. This is an indication of the bias present in the statistical model of order two, and that it may be advantageous to use a higher order model to approximate the objective within BOCS, as observed in Fig. 8 with greater N. While the order two model may be computationally efficient, future work will address adaptive switching to a higher order when there are enough training samples to estimate its parameters with low variance.

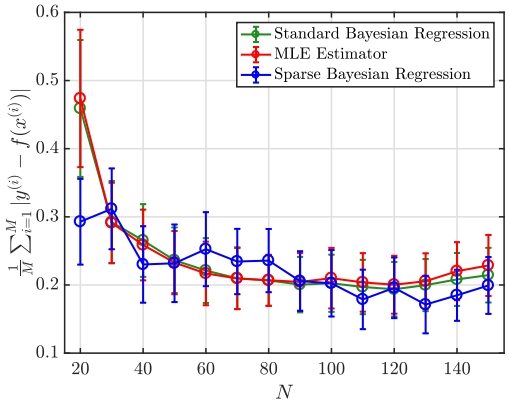

Figure 15. Test error of aero-structural problem for different $\boldsymbol{\alpha}$ estimators. The MLE (red), standard Bayesian linear regression (green) and sparse linear regression (blue) produce similar test error results.